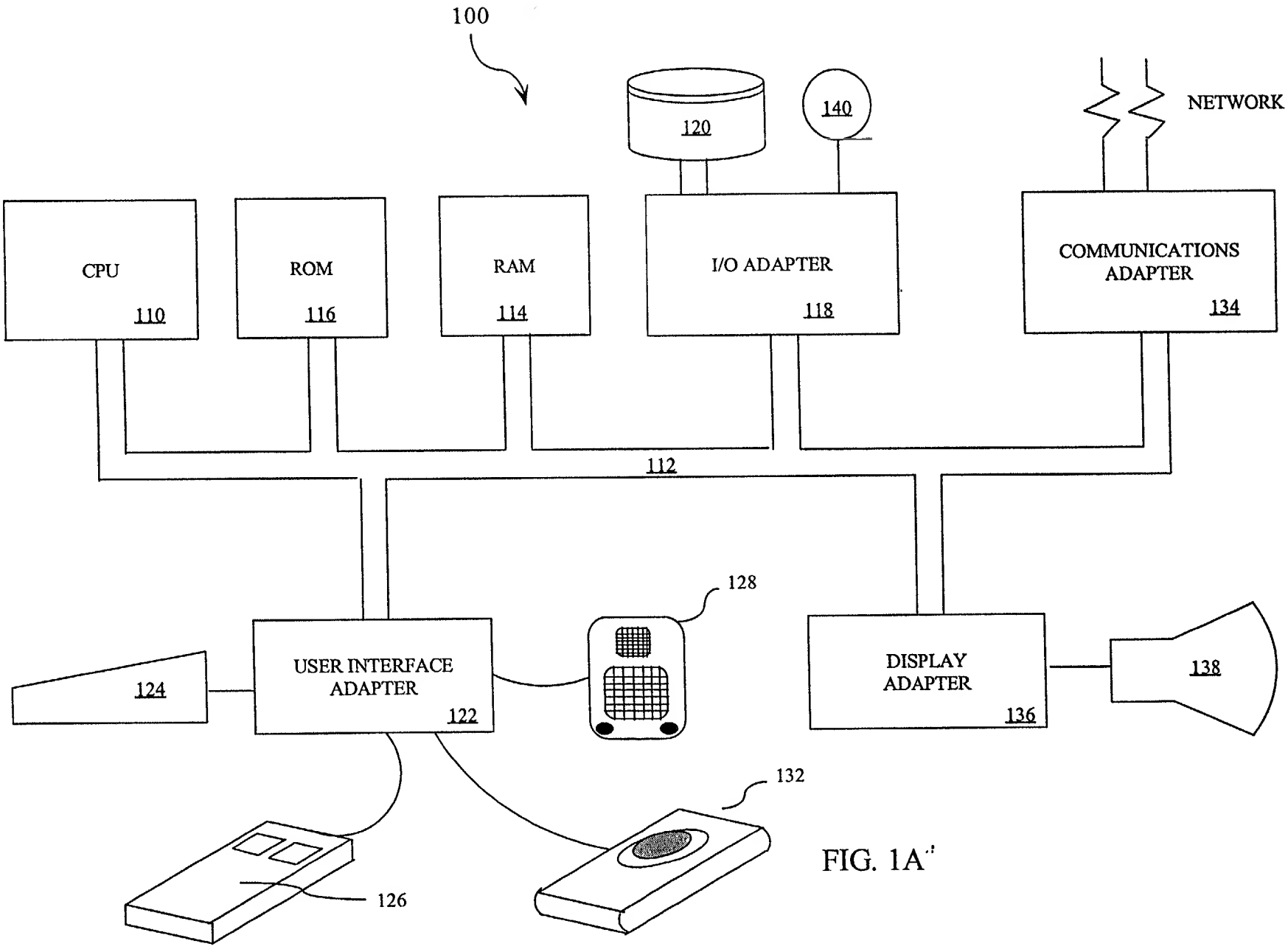

FIG. 1A

AT9-98-538 (ET 4010725US)

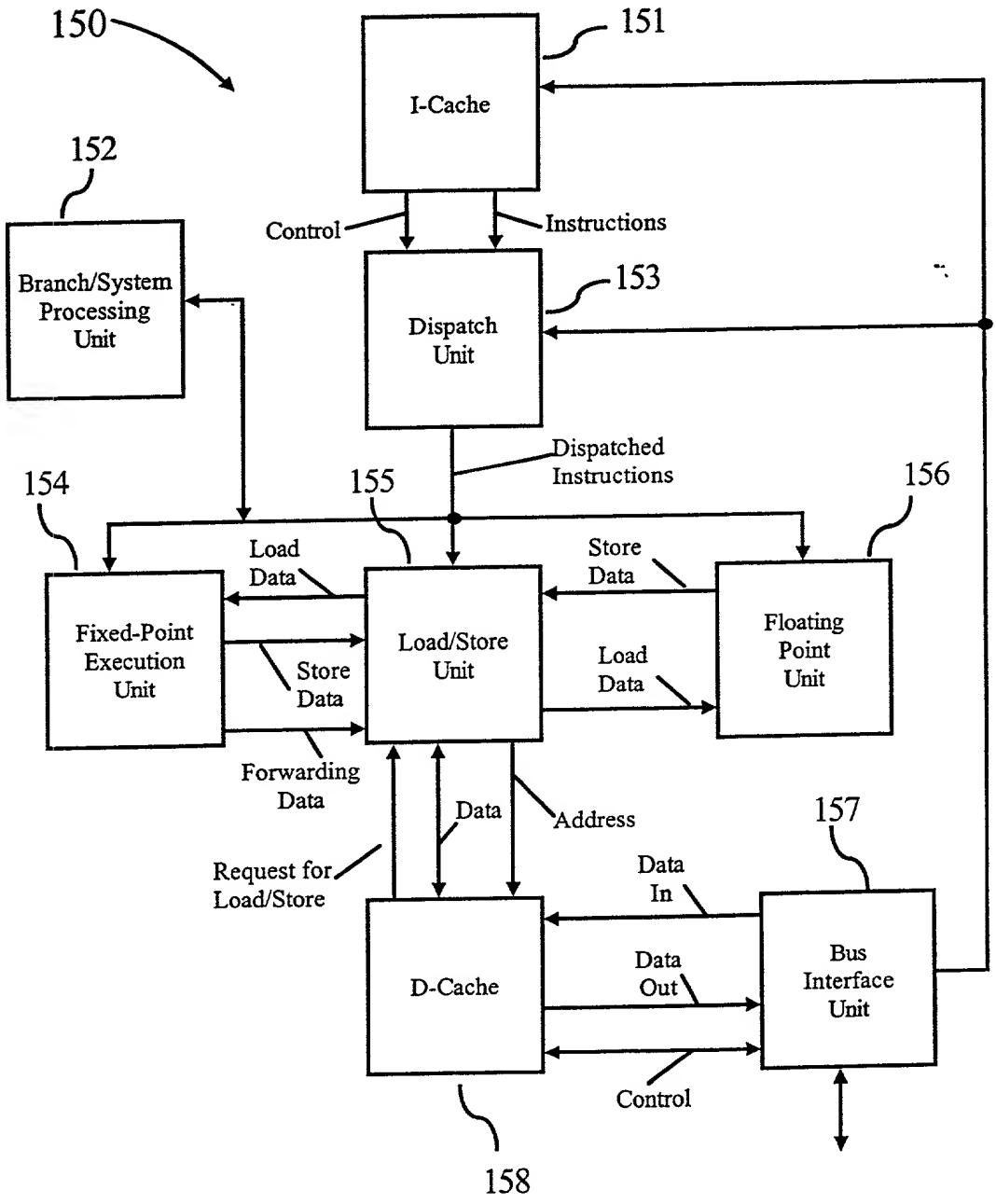
150
151
I-Cache
152
Branch/System Processing Unit
Control
Instructions
153
Dispatch Unit
Dispatched Instructions
154
155
156
Load Data
Store Data
Fixed-Point Execution Unit
Load/Store Unit
Floating Point Unit
Store Data
Load Data
Forwarding Data
Data
Address
157
Request for Load/Store
Data In
D-Cache
Data Out
Bus Interface Unit
Control
158

FIG. 1B

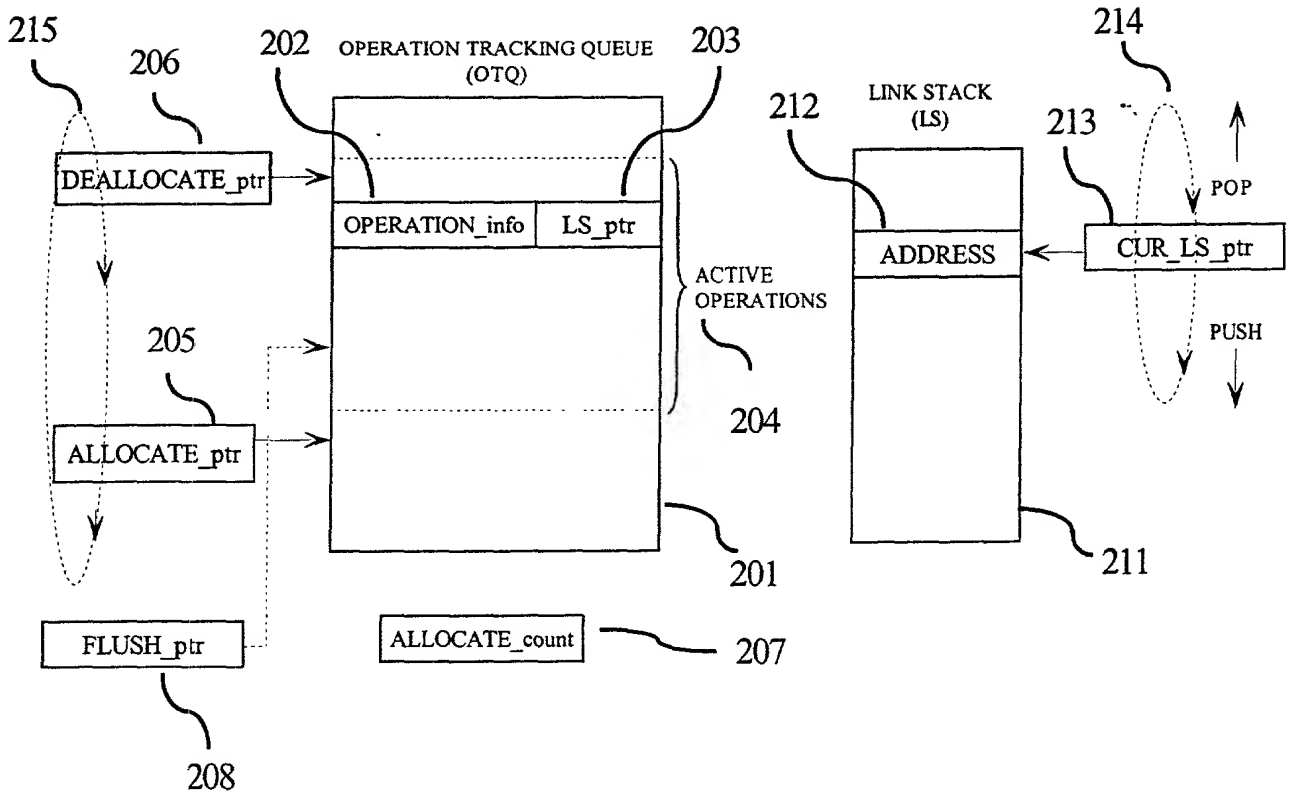
215
206
202
OPERATION TRACKING QUEUE
(OTQ)
203
214
212
LINK STACK
(LS)
213
DEALLOCATE_ptr
OPERATION_info
LS_ptr
POP
ADDRESS
CUR_LS_ptr
ACTIVE
OPERATIONS
205
PUSH
ALLOCATE_ptr
204
201
211
FLUSH_ptr
ALLOCATE_count
207
208

FIG. 2A

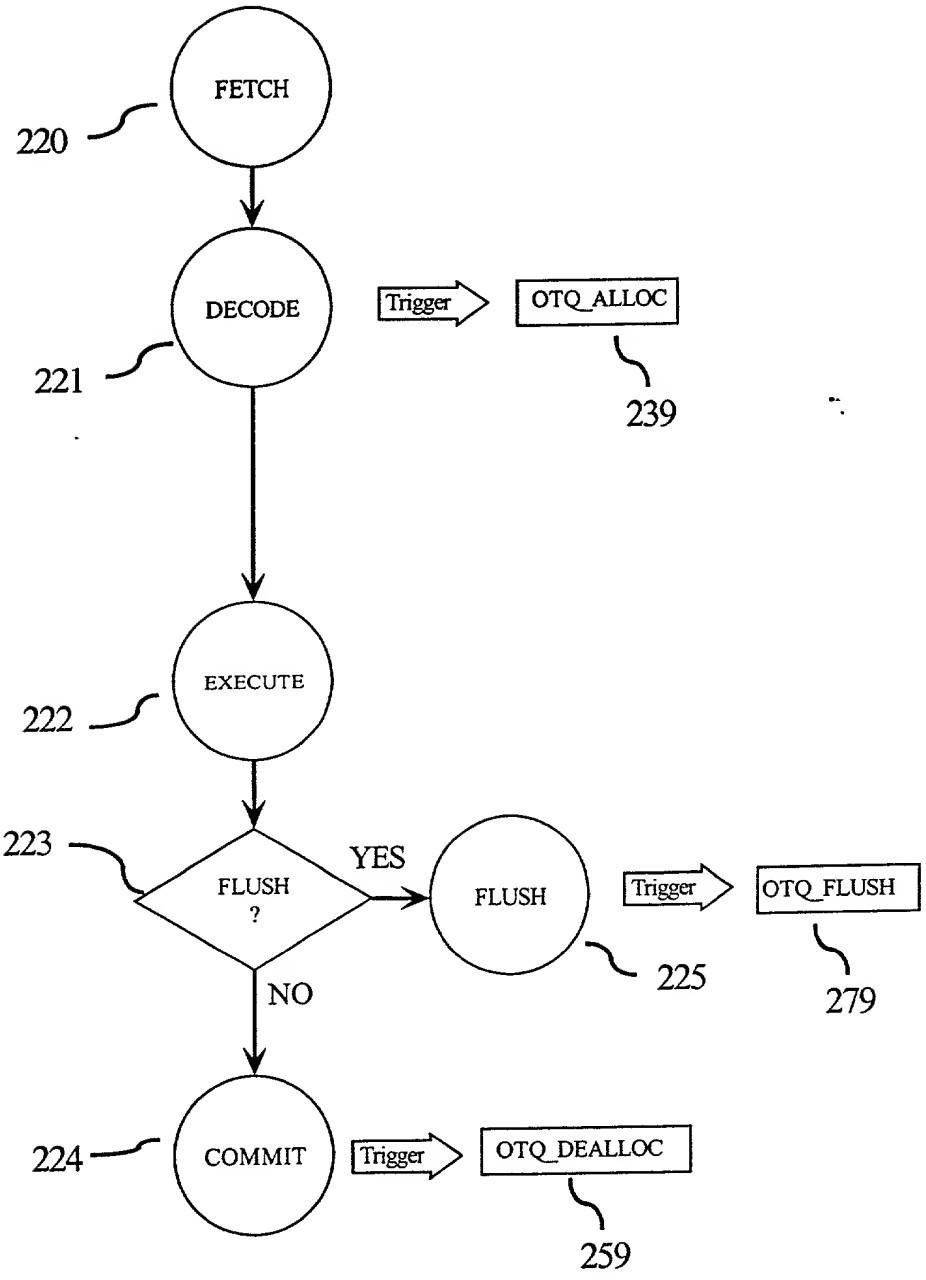
FETCH
220
DECODE
Trigger
OTQ_ALLOC
221
239
EXECUTE
222
223
FLUSH
?
YES
FLUSH
Trigger
OTQ_FLUSH
225
279
NO
224
COMMIT
Trigger
OTQ_DEALLOC
259

FIG. 2B

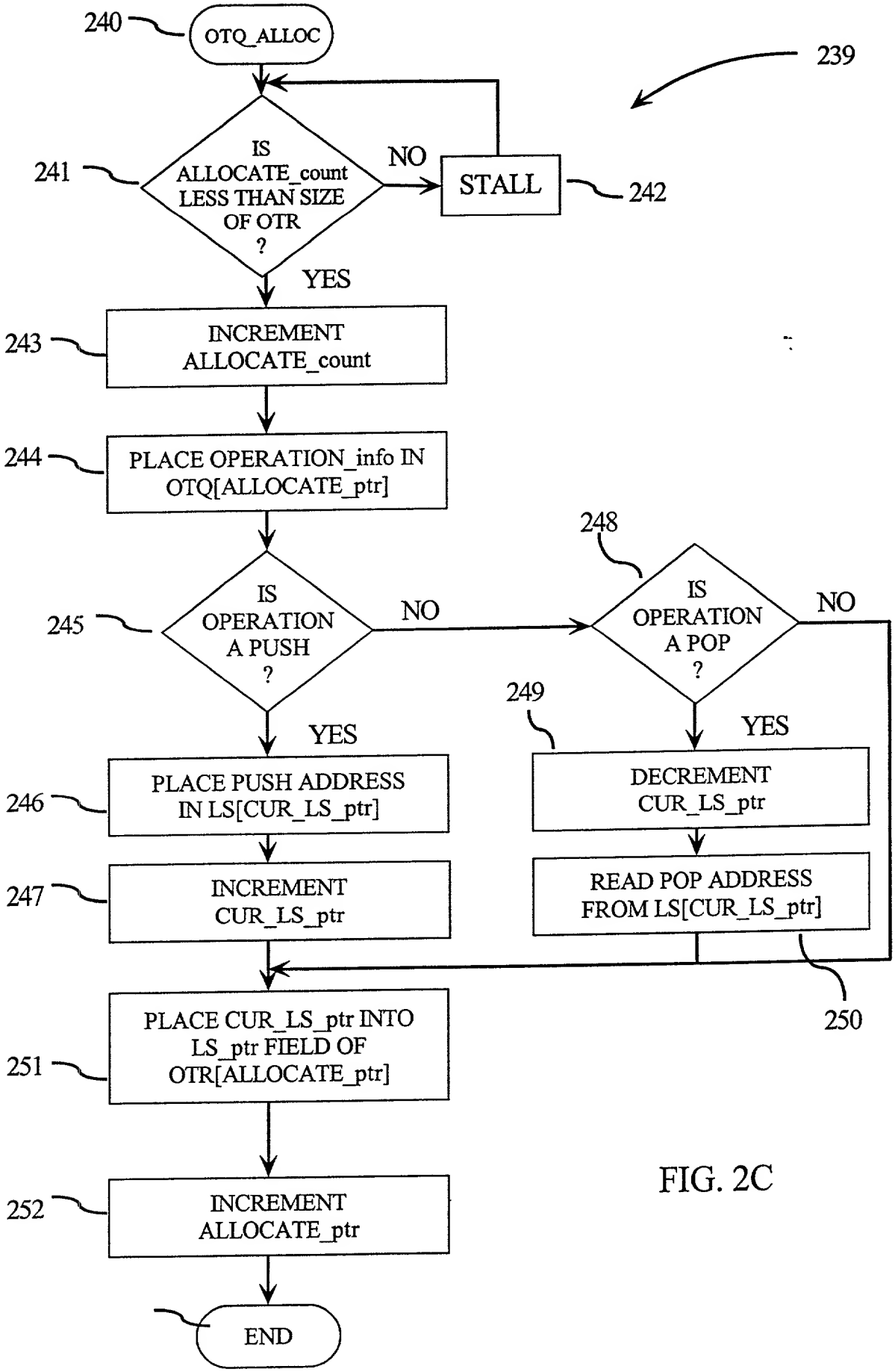
240
OTQ_ALLOC
239
241
IS ALLOCATE_count LESS THAN SIZE OF OTR ?
NO
STALL
242
YES
243
INCREMENT ALLOCATE_count
244
PLACE OPERATION_info IN OTQ[ALLOCATE_ptr]
248
245
IS OPERATION A PUSH ?
NO
IS OPERATION A POP ?
NO
249
YES
YES
246
PLACE PUSH ADDRESS IN LS[CUR_LS_ptr]
DECREMENT CUR_LS_ptr
247
INCREMENT CUR_LS_ptr
READ POP ADDRESS FROM LS[CUR_LS_ptr]
250
251
PLACE CUR_LS_ptr INTO LS_ptr FIELD OF OTR[ALLOCATE_ptr]
252
INCREMENT ALLOCATE_ptr
END

FIG. 2C

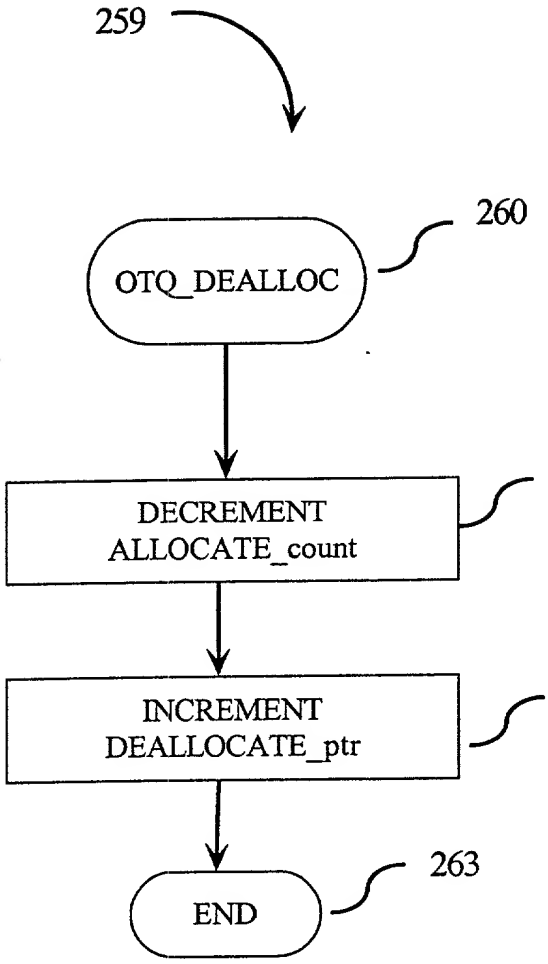
259
OTQ_DEALLOC
260
DECREMENT
ALLOCATE_count
261
INCREMENT
DEALLOCATE_ptr
262
END
263

FIG. 2D

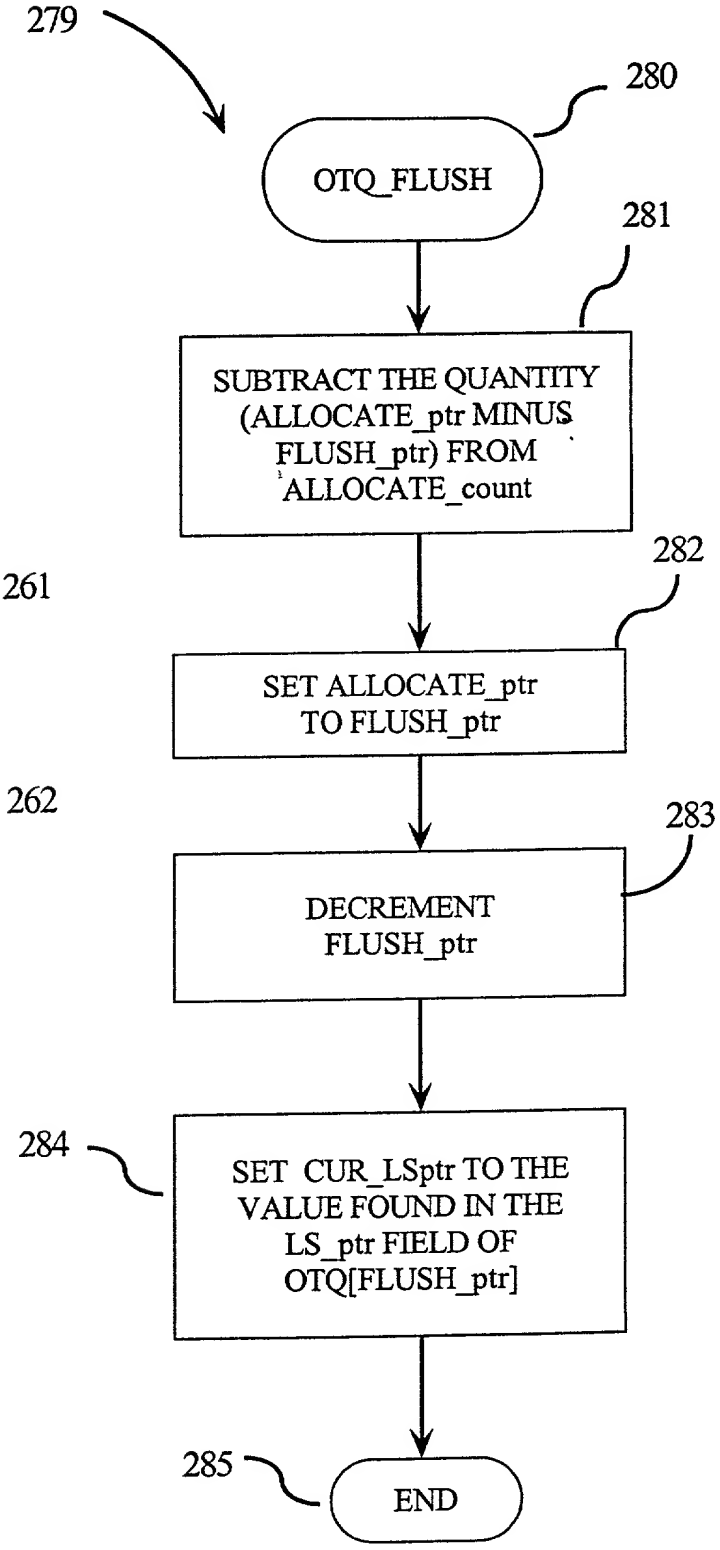
279
OTQ_FLUSH
280
SUBTRACT THE QUANTITY
(ALLOCATE_ptr MINUS
FLUSH_ptr) FROM
ALLOCATE_count
281
SET ALLOCATE_ptr
TO FLUSH_ptr
282
DECREMENT
FLUSH_ptr
283
SET CUR_LSptr TO THE
VALUE FOUND IN THE
LS_ptr FIELD OF
OTQ[FLUSH_ptr]
284
END
285

FIG. 2E

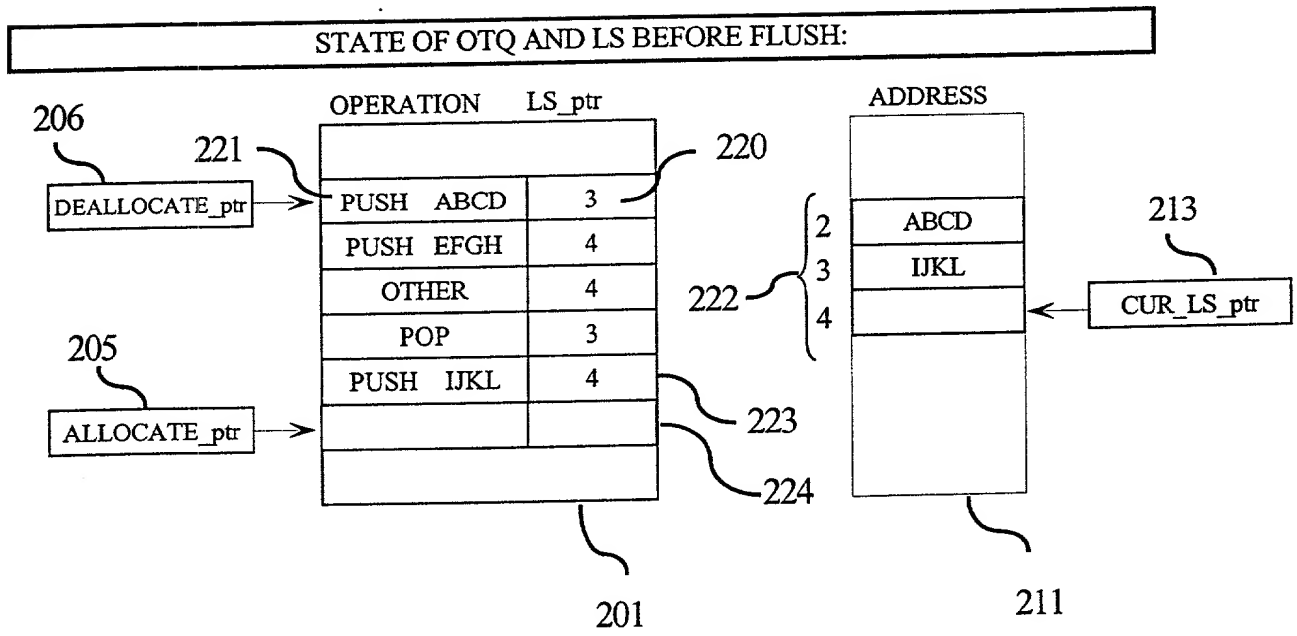
STATE OF OTQ AND LS BEFORE FLUSH:
OPERATION
LS_ptr
ADDRESS
206
221
220
DEALLOCATE_ptr
PUSH ABCD
3
PUSH EFGH
4
OTHER
4
POP
3
PUSH IJKL
4
205
ALLOCATE_ptr
223
224
201
2
3
4
222
ABCD
IJKL
213
CUR_LS_ptr
211

FIG. 2F

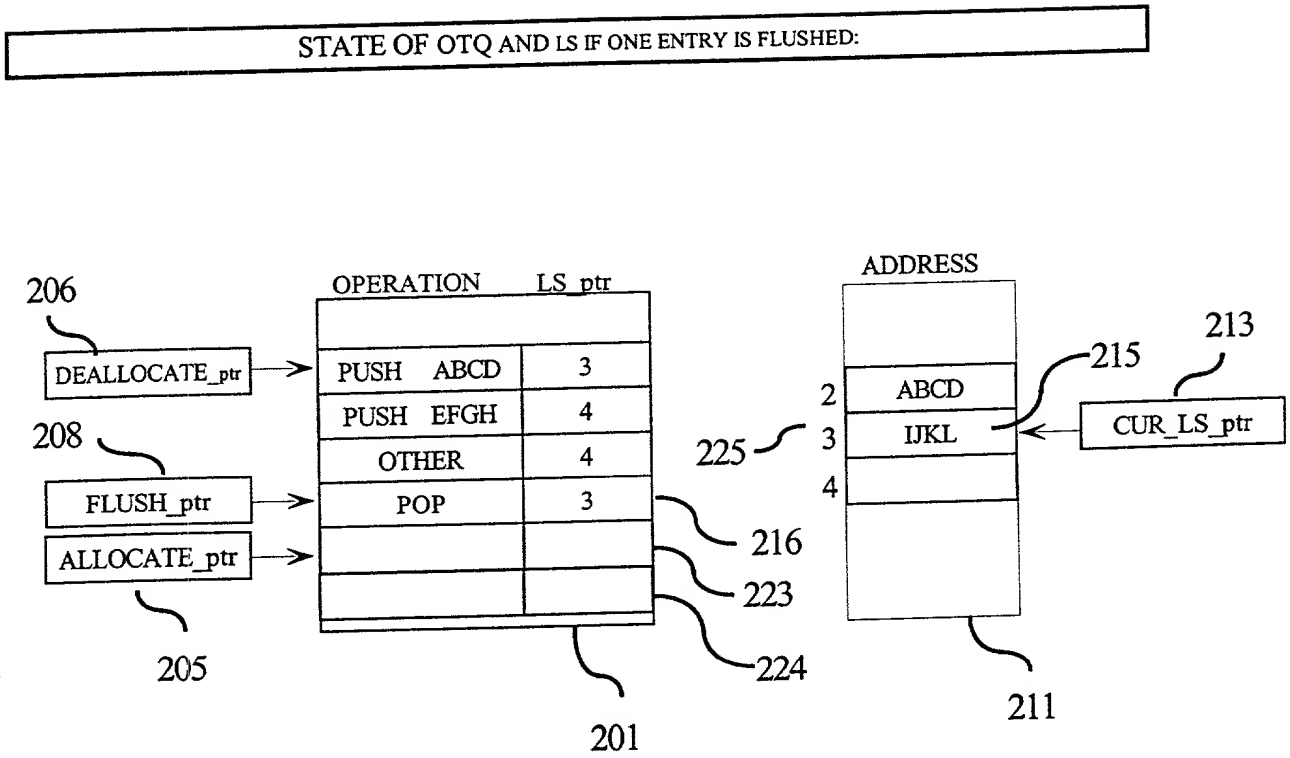
STATE OF OTQ AND LS IF ONE ENTRY IS FLUSHED:
206
DEALLOCATE_ptr
208
FLUSH_ptr
ALLOCATE_ptr
205
OPERATION
LS_ptr
PUSH ABCD
3
PUSH EFGH
4
OTHER
4
POP
3
201
216
223
224
225
ADDRESS
2
ABCD
3
IJKL
4
215
213
CUR_LS_ptr
211

FIG. 2G

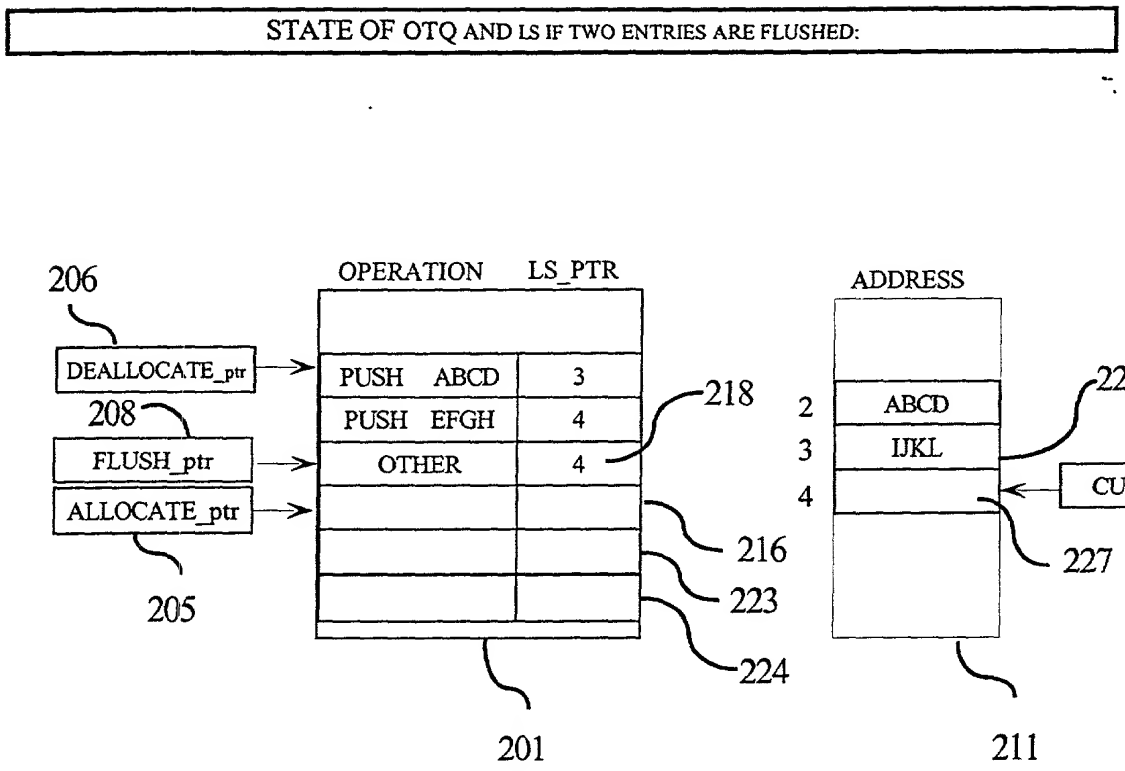
STATE OF OTQ AND LS IF TWO ENTRIES ARE FLUSHED:
206
DEALLOCATE_ptr
208
FLUSH_ptr
ALLOCATE_ptr
205
OPERATION
LS_PTR
PUSH ABCD
3
PUSH EFGH
4
OTHER
4
218
216
223
224
201
ADDRESS
2
ABCD
3
IJKL
4
228
213
CUR_LS_ptr
227
211

FIG. 2H

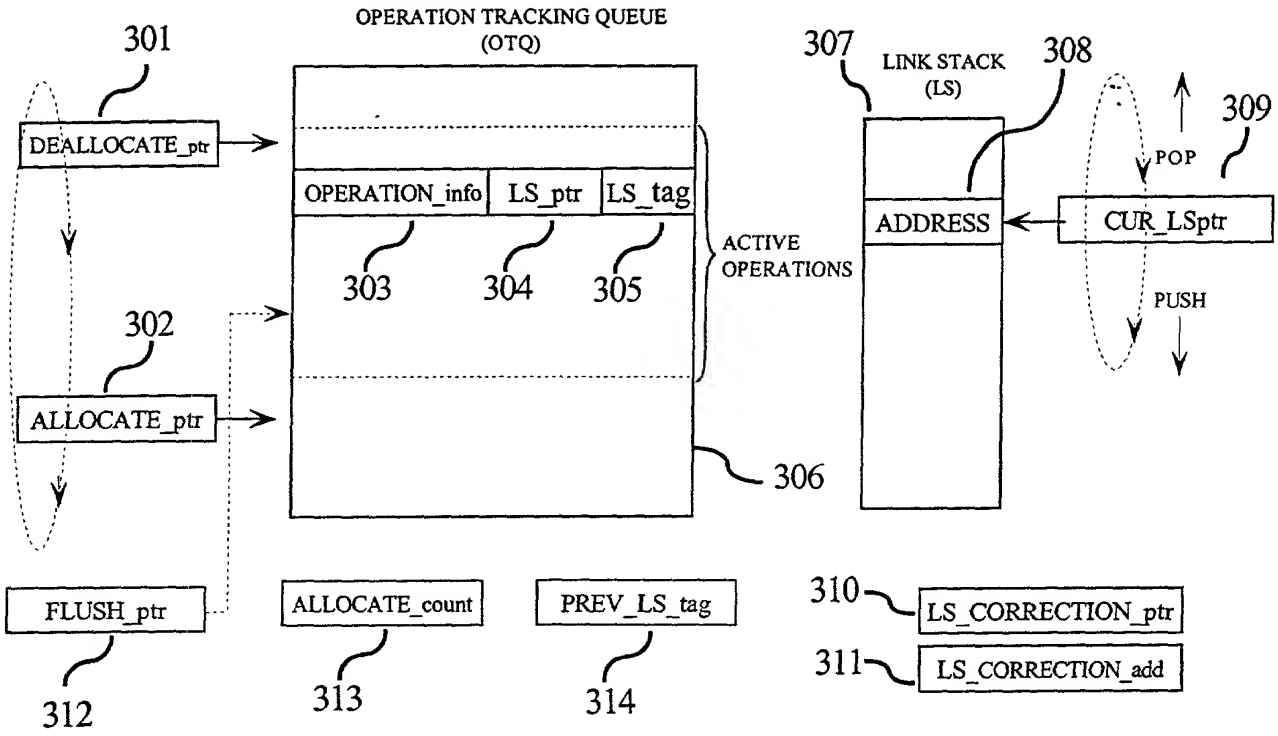
OPERATION TRACKING QUEUE
(OTQ)
301
DEALLOCATE_ptr
OPERATION_info
LS_ptr
LS_tag
303
304
305
ACTIVE
OPERATIONS
302
ALLOCATE_ptr
306
307
LINK STACK
(LS)
308
ADDRESS
309
POP
CUR_LSptr
PUSH
FLUSH_ptr
312
ALLOCATE_count
313
PREV_LS_tag
314
310
LS_CORRECTION_ptr
311
LS_CORRECTION_add

FIG. 3A

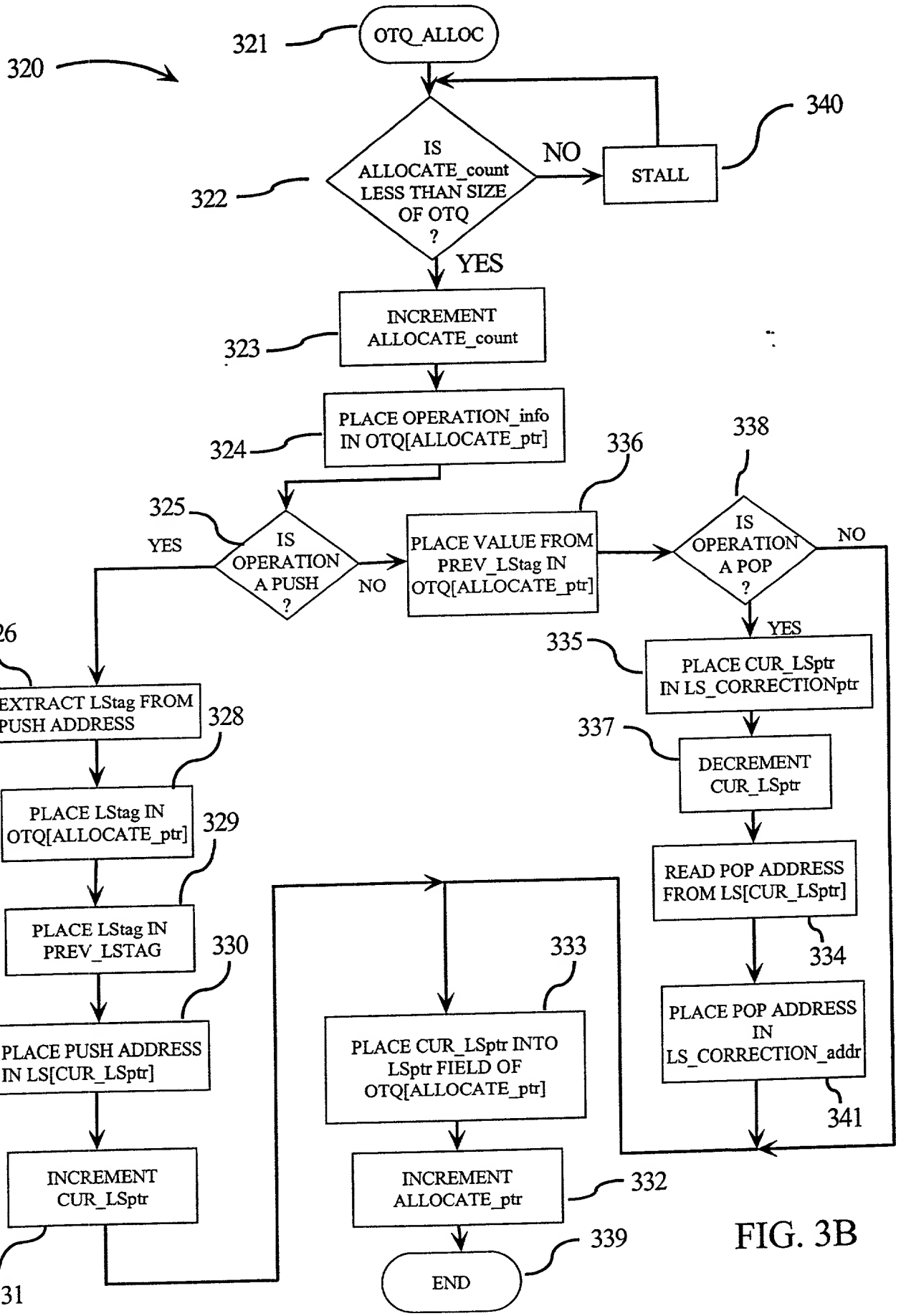
320
321
OTQ_ALLOC
IS ALLOCATE_count LESS THAN SIZE OF OTQ ?
322
NO
STALL
340
YES
INCREMENT ALLOCATE_count
323
PLACE OPERATION_info IN OTQ[ALLOCATE_ptr]
324
325
IS OPERATION A PUSH ?
YES
NO
PLACE VALUE FROM PREV_LStag IN OTQ[ALLOCATE_ptr]
336
338
IS OPERATION A POP ?
NO
YES
326
EXTRACT LStag FROM PUSH ADDRESS
328
PLACE LStag IN OTQ[ALLOCATE_ptr]
329
PLACE LStag IN PREV_LSTAG
330
PLACE PUSH ADDRESS IN LS[CUR_LSptr]
INCREMENT CUR_LSptr
331
335
PLACE CUR_LSptr IN LS_CORRECTIONptr
337
DECREMENT CUR_LSptr
READ POP ADDRESS FROM LS[CUR_LSptr]
334
PLACE POP ADDRESS IN LS_CORRECTION_addr
341
333
PLACE CUR_LSptr INTO LSptr FIELD OF OTQ[ALLOCATE_ptr]
INCREMENT ALLOCATE_ptr
332
339
END

FIG. 3B

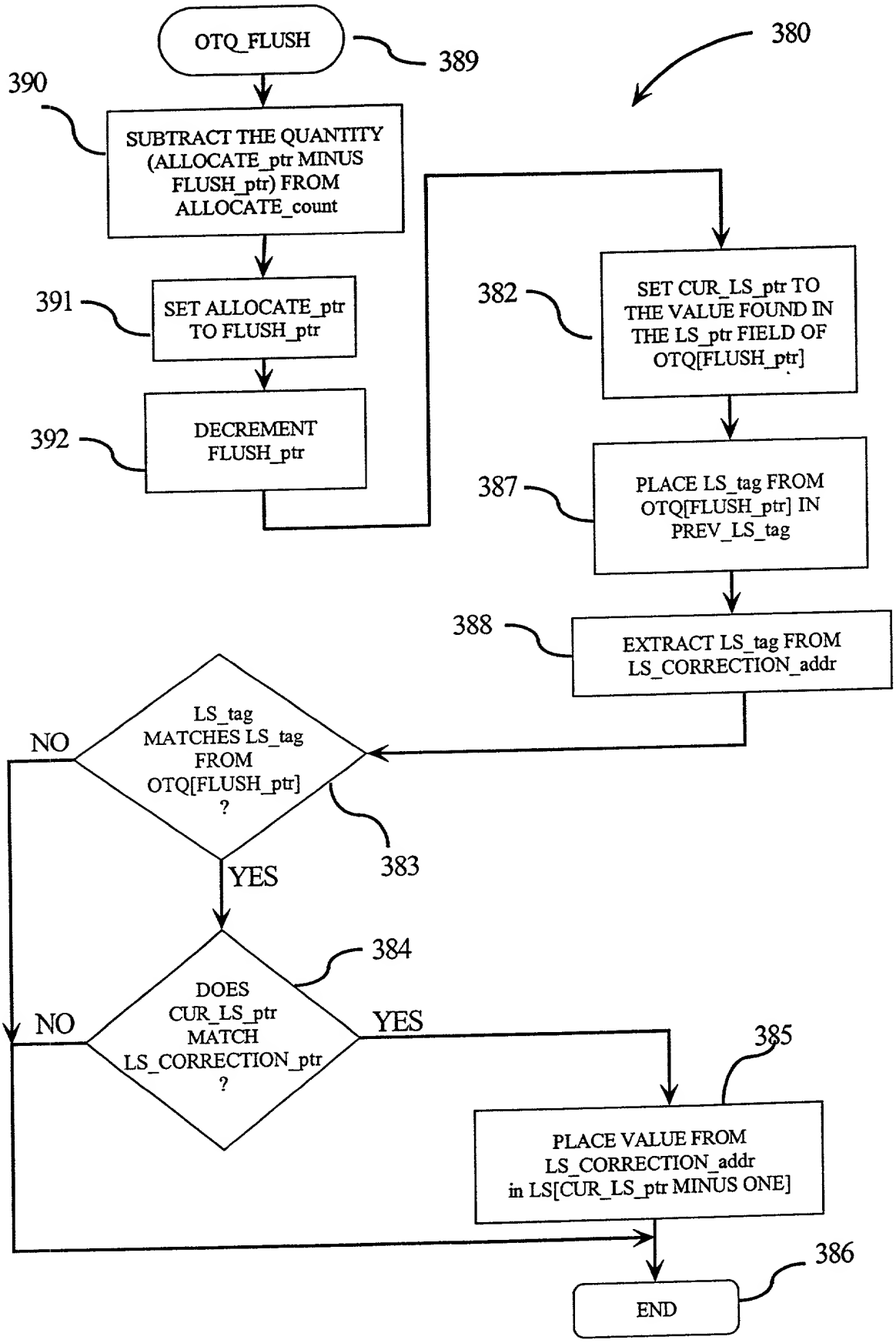
OTQ_FLUSH
389
380
390
SUBTRACT THE QUANTITY (ALLOCATE_ptr MINUS FLUSH_ptr) FROM ALLOCATE_count
391
SET ALLOCATE_ptr TO FLUSH_ptr
392
DECREMENT FLUSH_ptr
382
SET CUR_LS_ptr TO THE VALUE FOUND IN THE LS_ptr FIELD OF OTQ[FLUSH_ptr]
387
PLACE LS_tag FROM OTQ[FLUSH_ptr] IN PREV_LS_tag
388
EXTRACT LS_tag FROM LS_CORRECTION_addr
LS_tag MATCHES LS_tag FROM OTQ[FLUSH_ptr] ?
NO
YES
383
384
DOES CUR_LS_ptr MATCH LS_CORRECTION_ptr ?
NO
YES
385
PLACE VALUE FROM LS_CORRECTION_addr in LS[CUR_LS_ptr MINUS ONE]
386
END

FIG. 3C

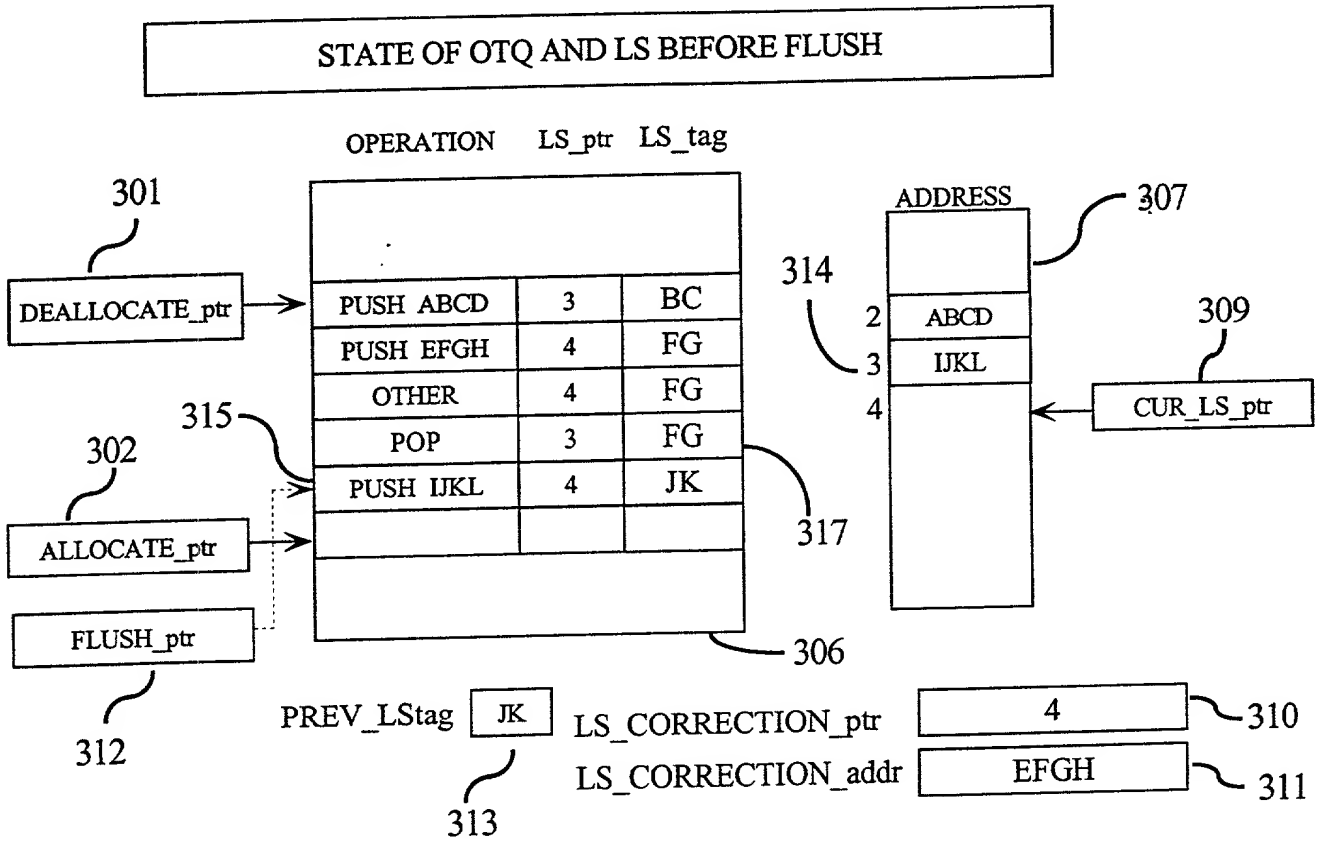
STATE OF OTQ AND LS BEFORE FLUSH
OPERATION
LS_ptr
LS_tag
301
DEALLOCATE_ptr
PUSH ABCD 3 BC
PUSH EFGH 4 FG
OTHER 4 FG
POP 3 FG
PUSH IJKL 4 JK
315
302
ALLOCATE_ptr
FLUSH_ptr
312
317
306
ADDRESS
307
314
2 ABCD
3 IJKL
4
309
CUR_LS_ptr
PREV_LStag
JK
313
LS_CORRECTION_ptr
4
310
LS_CORRECTION_addr
EFGH
311

FIG. 3D

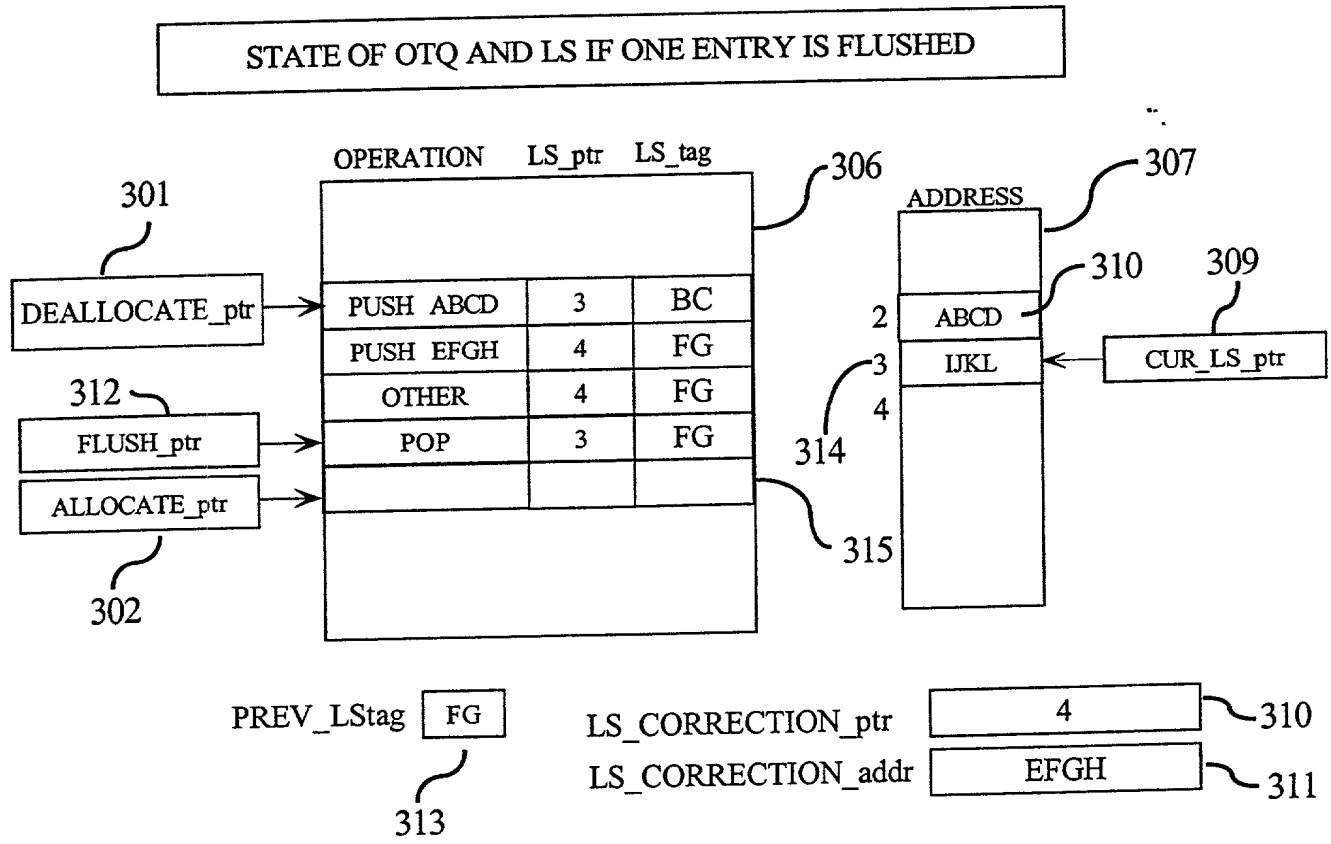
STATE OF OTQ AND LS IF ONE ENTRY IS FLUSHED
OPERATION
LS_ptr
LS_tag
306
ADDRESS
307
301
DEALLOCATE_ptr
PUSH ABCD
3
BC
PUSH EFGH
4
FG
OTHER
4
FG
POP
3
FG
310
309
2
ABCD
IJKL
CUR_LS_ptr
3
4
314
312
FLUSH_ptr
ALLOCATE_ptr
315
302
PREV_LStag
FG
313
LS_CORRECTION_ptr
4
310
LS_CORRECTION_addr
EFGH
311

FIG. 3E

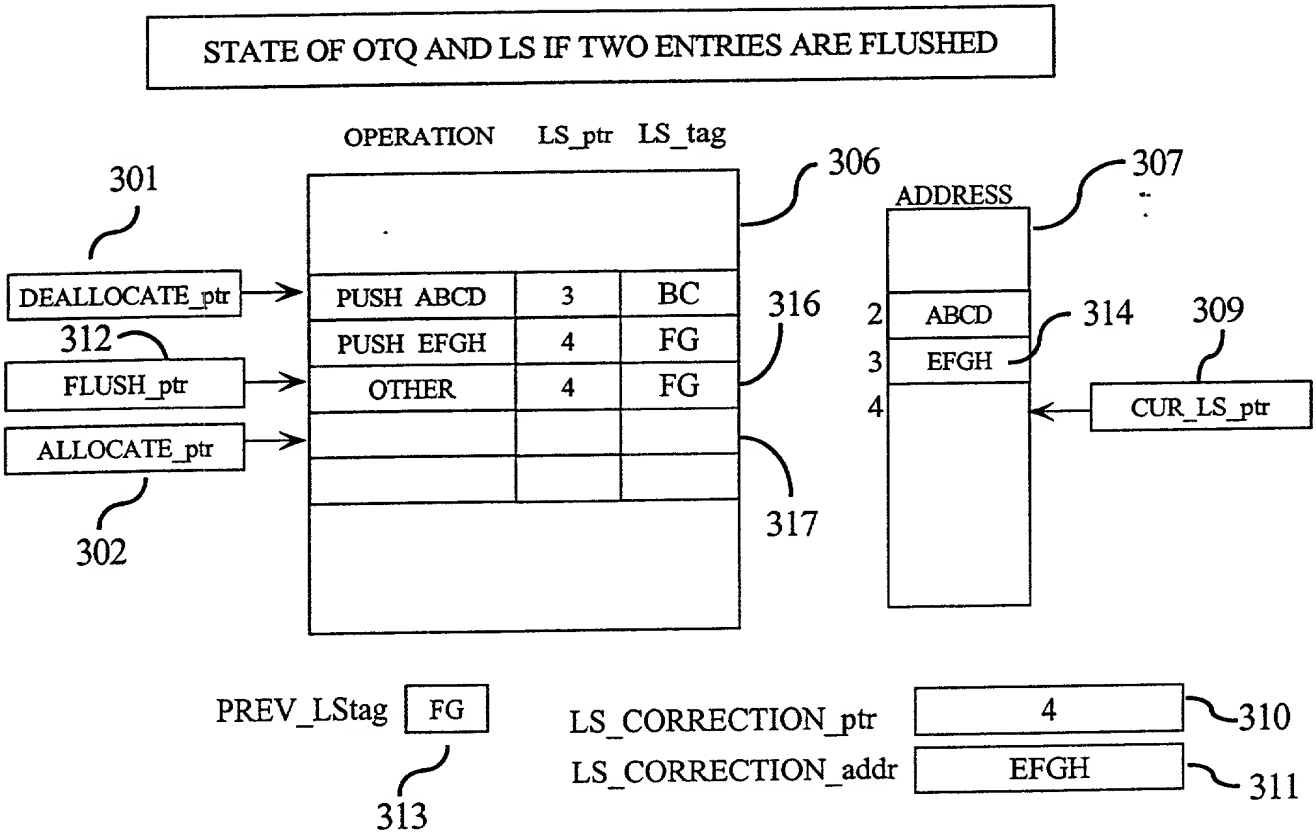
STATE OF OTQ AND LS IF TWO ENTRIES ARE FLUSHED
OPERATION
LS_ptr
LS_tag
301
306
307
ADDRESS
DEALLOCATE_ptr
PUSH ABCD
3
BC
316
2
ABCD
314
309
312
FLUSH_ptr
PUSH EFGH
4
FG
3
EFGH
OTHER
4
FG
4
CUR_LS_ptr
ALLOCATE_ptr
317
302
PREV_LStag
FG
LS_CORRECTION_ptr
4
310
LS_CORRECTION_addr
EFGH
311
313

FIG. 3F

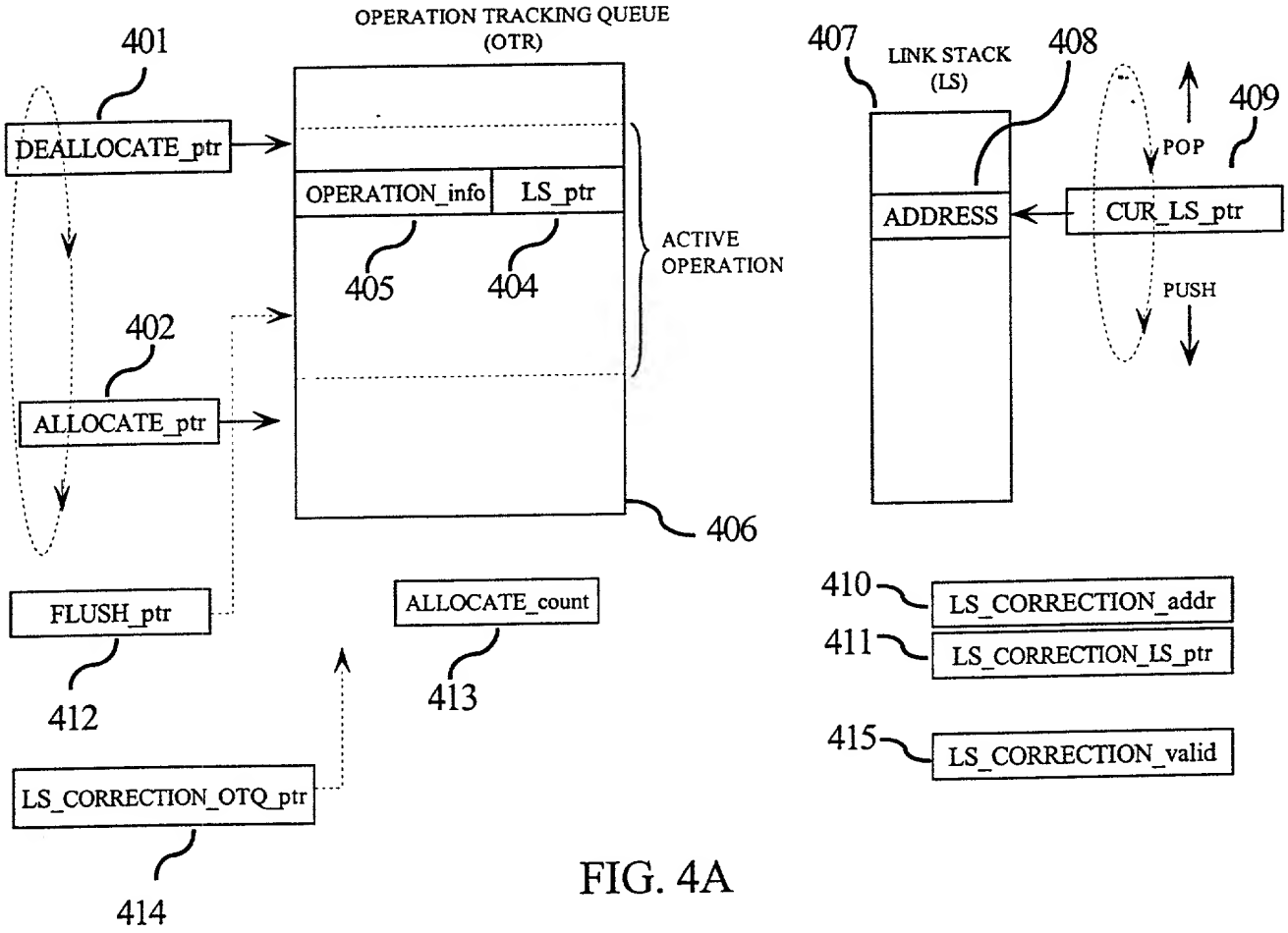
OPERATION TRACKING QUEUE
(OTR)
401
DEALLOCATE_ptr
OPERATION_info
LS_ptr
405
404
ACTIVE
OPERATION
406
402
ALLOCATE_ptr
FLUSH_ptr
412
ALLOCATE_count
413
LS_CORRECTION_OTQ_ptr
414
407
LINK STACK
(LS)
408
409
POP
ADDRESS
CUR_LS_ptr
PUSH
410
LS_CORRECTION_addr
411
LS_CORRECTION_LS_ptr
415
LS_CORRECTION_valid

FIG. 4A

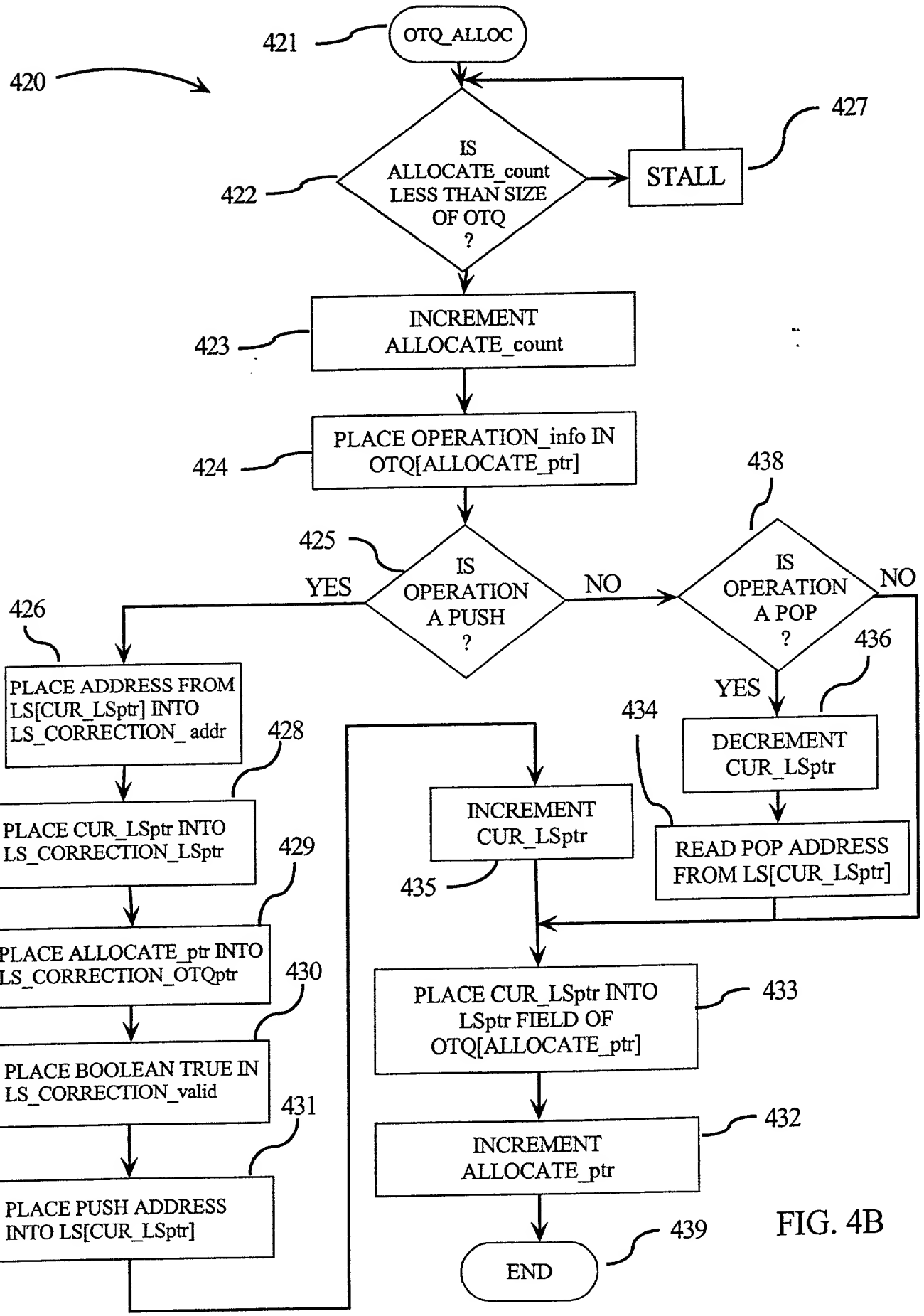
420
421
OTQ_ALLOC
422
IS ALLOCATE_count LESS THAN SIZE OF OTQ ?
427
STALL
423
INCREMENT ALLOCATE_count
424
PLACE OPERATION_info IN OTQ[ALLOCATE_ptr]
425
IS OPERATION A PUSH ?
YES
NO
438
IS OPERATION A POP ?
NO
426
PLACE ADDRESS FROM LS[CUR_LSptr] INTO LS_CORRECTION_ addr
428
PLACE CUR_LSptr INTO LS_CORRECTION_LSptr
429
PLACE ALLOCATE_ptr INTO LS_CORRECTION_OTQptr
430
PLACE BOOLEAN TRUE IN LS_CORRECTION_valid
431
PLACE PUSH ADDRESS INTO LS[CUR_LSptr]
YES
436
DECREMENT CUR_LSptr
434
READ POP ADDRESS FROM LS[CUR_LSptr]
INCREMENT CUR_LSptr
435
PLACE CUR_LSptr INTO LSptr FIELD OF OTQ[ALLOCATE_ptr]
433
INCREMENT ALLOCATE_ptr
432
END
439

FIG. 4B

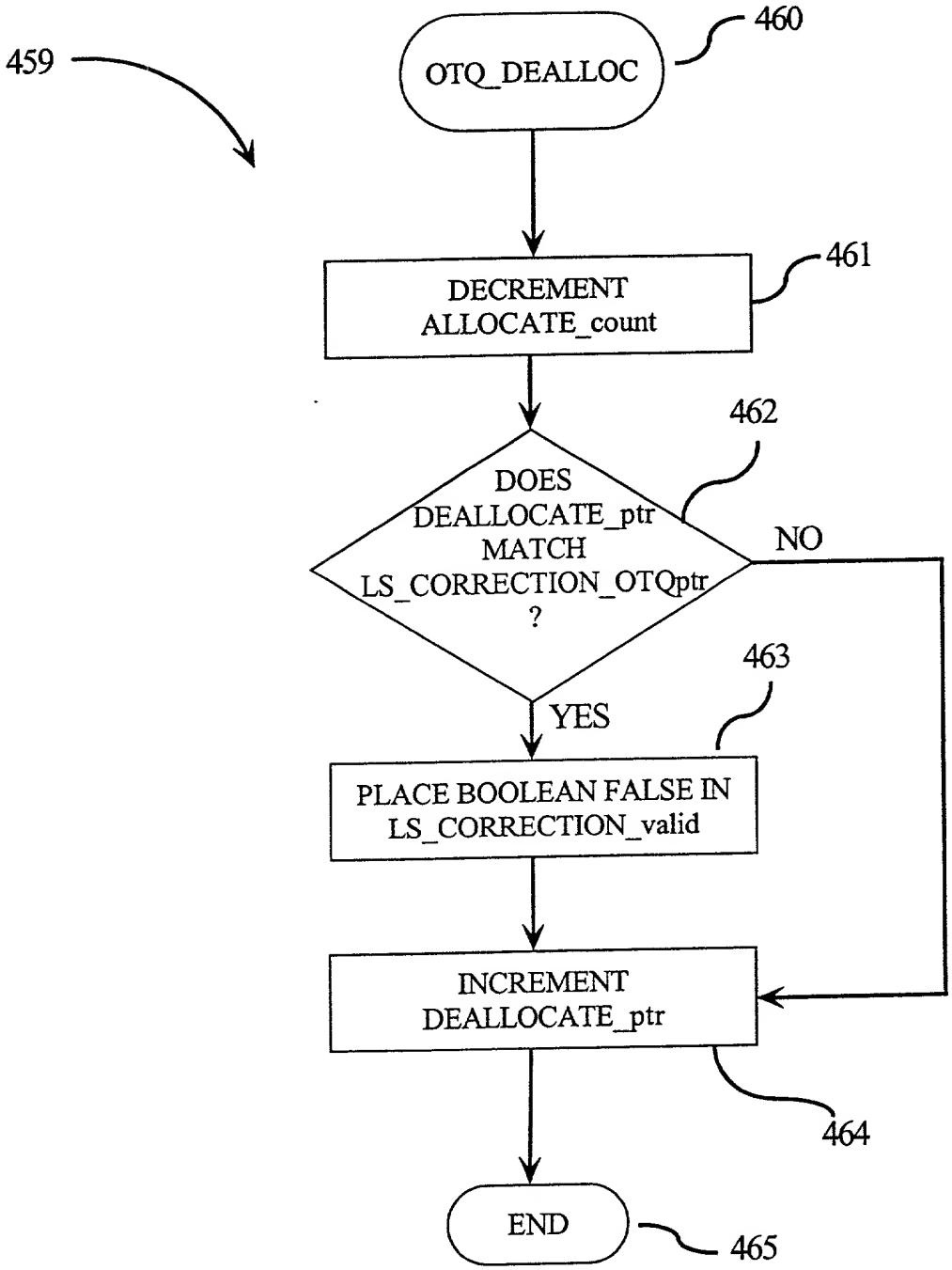
459
OTQ_DEALLOC
460
DECREMENT
ALLOCATE_count
461
DOES
DEALLOCATE_ptr
MATCH
LS_CORRECTION_OTQptr
?
462
NO
YES
463
PLACE BOOLEAN FALSE IN
LS_CORRECTION_valid
INCREMENT
DEALLOCATE_ptr
464
END
465

FIG. 4C

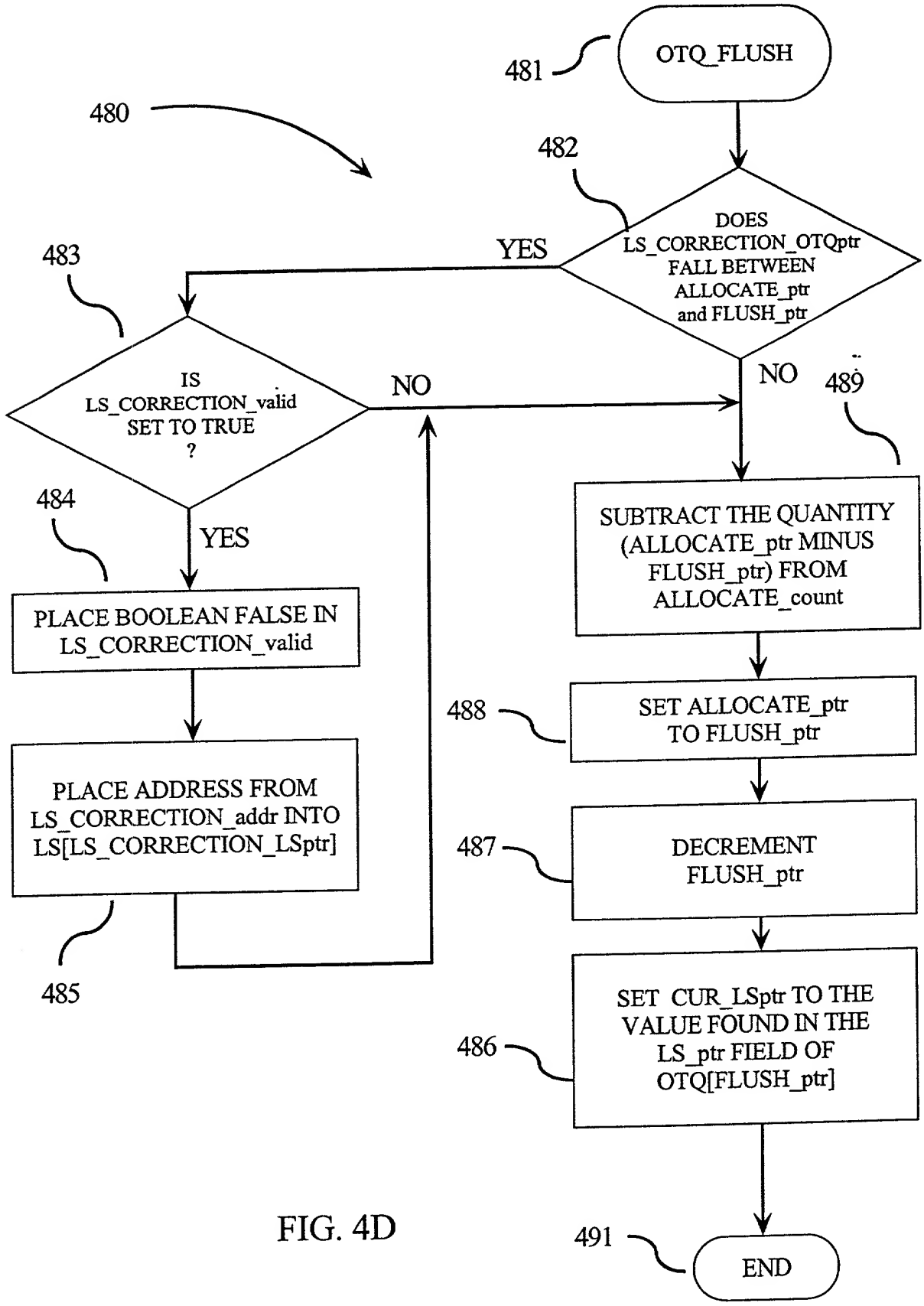
480
481
OTQ_FLUSH
482
DOES LS_CORRECTION_OTQptr FALL BETWEEN ALLOCATE_ptr and FLUSH_ptr
YES
NO
483
IS LS_CORRECTION_valid SET TO TRUE ?
NO
489
484
YES
PLACE BOOLEAN FALSE IN LS_CORRECTION_valid
SUBTRACT THE QUANTITY (ALLOCATE_ptr MINUS FLUSH_ptr) FROM ALLOCATE_count
488
SET ALLOCATE_ptr TO FLUSH_ptr
PLACE ADDRESS FROM LS_CORRECTION_addr INTO LS[LS_CORRECTION_LSptr]
487
DECREMENT FLUSH_ptr
SET CUR_LSptr TO THE VALUE FOUND IN THE LS_ptr FIELD OF OTQ[FLUSH_ptr]
486
485
491
END

FIG. 4D

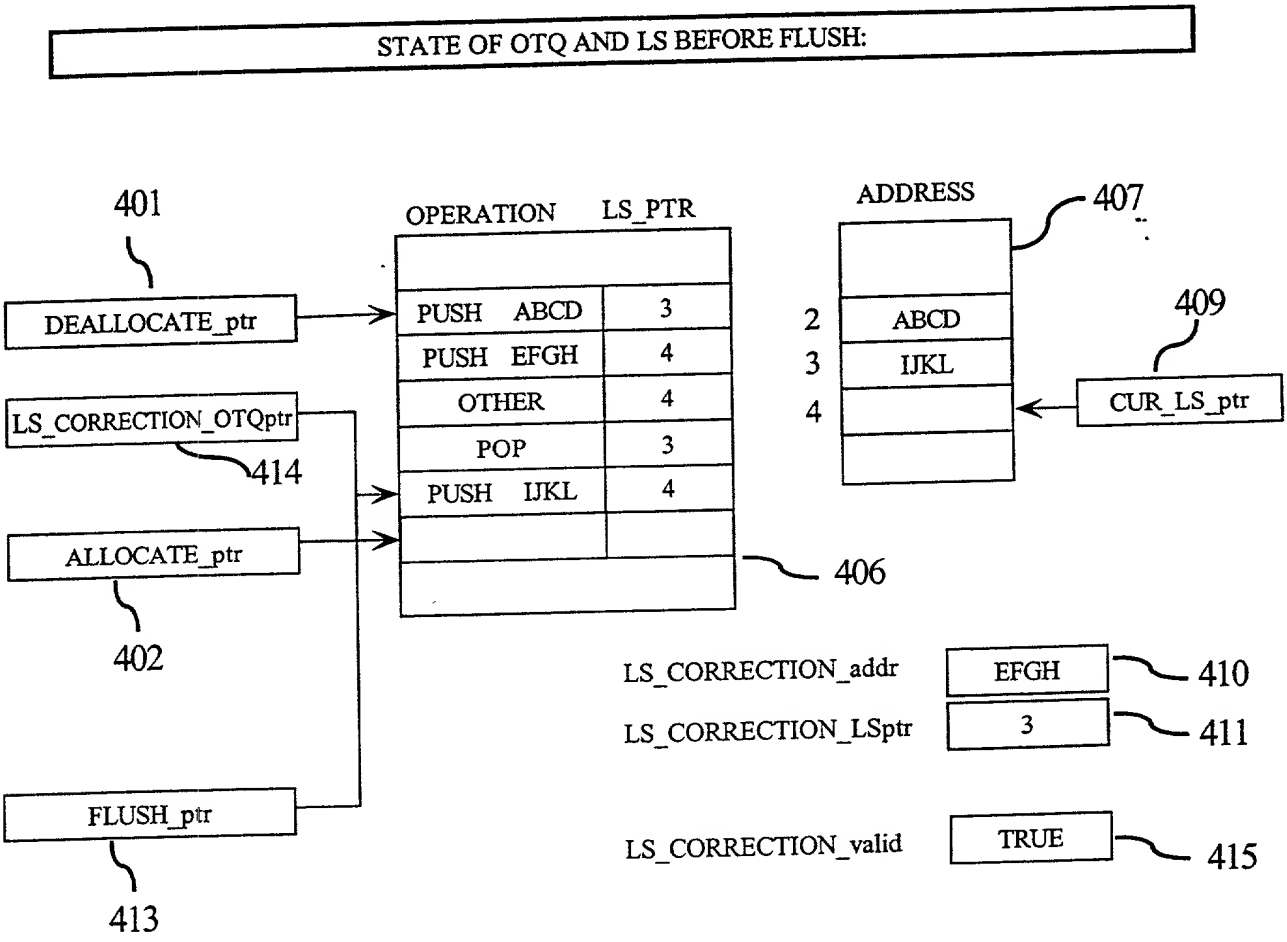
STATE OF OTQ AND LS BEFORE FLUSH:
401
DEALLOCATE_ptr
LS_CORRECTION_OTQptr
414
ALLOCATE_ptr
402
FLUSH_ptr
413
OPERATION
LS_PTR
PUSH ABCD
3
PUSH EFGH
4
OTHER
4
POP
3
PUSH IJKL
4
406
ADDRESS
407
2
ABCD
3
IJKL
4
409
CUR_LS_ptr
LS_CORRECTION_addr
EFGH
410
LS_CORRECTION_LSptr
3
411
LS_CORRECTION_valid
TRUE
415

FIG. 4E

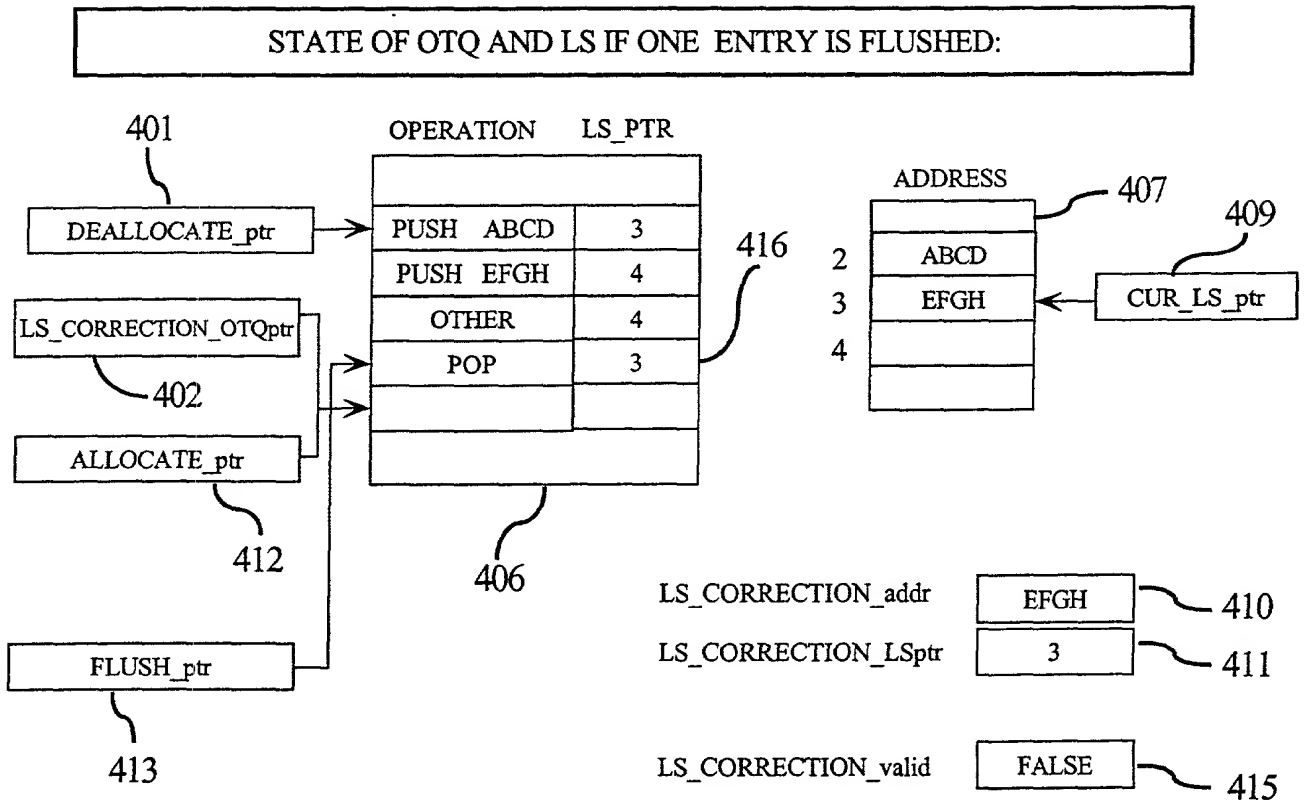
STATE OF OTQ AND LS IF ONE ENTRY IS FLUSHED:
401
OPERATION
LS_PTR
DEALLOCATE_ptr
PUSH ABCD
3
PUSH EFGH
4
OTHER
4
POP
3
416
LS_CORRECTION_OTQptr
402
ALLOCATE_ptr
412
406
ADDRESS
407
409
2
ABCD
3
EFGH
CUR_LS_ptr
4
FLUSH_ptr
413
LS_CORRECTION_addr
EFGH
410
LS_CORRECTION_LSptr
3
411
LS_CORRECTION_valid
FALSE
415

FIG. 4F

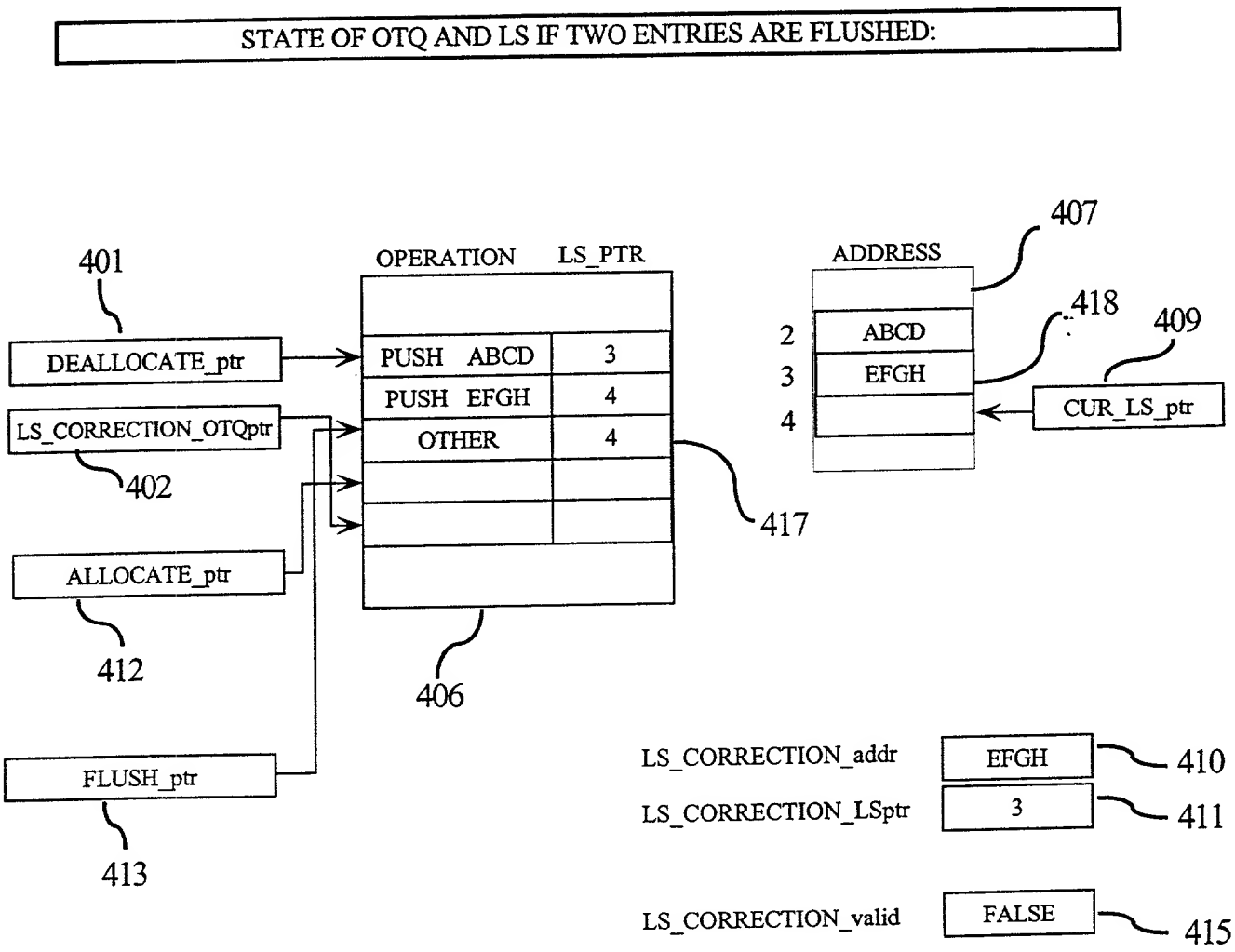
STATE OF OTQ AND LS IF TWO ENTRIES ARE FLUSHED:
401
DEALLOCATE_ptr
LS_CORRECTION_OTQptr
402
ALLOCATE_ptr
412
FLUSH_ptr
413
OPERATION
LS_PTR
PUSH ABCD
3
PUSH EFGH
4
OTHER
4
406
417
ADDRESS
407
2
ABCD
3
EFGH
4
418
409
CUR_LS_ptr
LS_CORRECTION_addr
EFGH
410
LS_CORRECTION_LSptr
3
411
LS_CORRECTION_valid
FALSE
415

FIG. 4G